IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A computer-implemented method for ~~programmatically~~ automatically generating a graphical program based on a state diagram, comprising:

receiving state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states;

~~programmatically~~ automatically generating the graphical program in response to the state diagram information, wherein said ~~programmatically~~ automatically generating comprises ~~programmatically~~ automatically generating graphical source code corresponding to the plurality of states, wherein the graphical source code comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, [[and]] wherein the graphical program is executable by a computer, and wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

2. (Cancelled)

3. (Previously Presented) The method of claim 1,
wherein the state diagram represents desired operation of a software program.

4. (Previously Presented) The method of claim 1,
wherein the state diagram represents desired operation of a hardware device.

5. (Previously Presented) The method of claim 1,
wherein the state diagram represents a desired algorithm.

6. (Previously Presented) The method of claim 1,
wherein the state diagram represents a test sequence.

7. (Cancelled)

8. (Currently Amended) The method of claim 1,

wherein said ~~programmatically~~ automatically generating the graphical program comprises ~~programmatically~~ automatically generating a block diagram including the graphical source code corresponding to the specified ~~one or more~~ plurality of states.

9-11. (Cancelled)

12. (Currently Amended) The method of claim 1,

wherein, for at least one state of the plurality of states, the state diagram information specifies program code associated with the state;

wherein the ~~programmatically~~ automatically generated graphical source code includes the specified program code.

13. (Currently Amended) The method of claim 1,

wherein, for at least one state, the state diagram information specifies program code associated with the state;

wherein the ~~programmatically~~ automatically generated graphical source code is operable to invoke the specified source code.

14. (Currently Amended) The method of claim 1,

wherein the state diagram information further specifies one or more state transitions, wherein each state transition specifies a transition from a first state to a second state;

wherein said ~~programmatically~~ automatically generating further comprises ~~programmatically~~ automatically generating graphical source code corresponding to the specified state transitions.

15. (Currently Amended) The method of claim 14,

wherein the ~~programmatically~~ <u>automatically</u> generated graphical source code includes placeholder graphical source code for each state transition.

16. (Original) The method of claim 15, further comprising:

for one or more state transitions, a user manually entering graphical source code specifying a Boolean condition associated with the state transition.

17. (Currently Amended) The method of claim 14,

wherein the state diagram information specifies at least two state transitions from a particular state;

wherein the state diagram information also specifies a priority ordering for the at least two state transitions;

wherein said ~~programmatically~~ <u>automatically</u> generating comprises ~~programmatically~~ <u>automatically</u> generating graphical source code such that, during execution of the graphical program, Boolean conditions associated with the at least two state transitions are evaluated in the specified priority ordering.

18. (Currently Amended) The method of claim 1,

wherein the state diagram information further specifies an initially active state;

wherein said ~~programmatically~~ <u>automatically</u> generating comprises ~~programmatically~~ <u>automatically</u> generating graphical source code such that the graphical program begins execution in the initially active state.

19. (Currently Amended) The method of claim 1,

wherein the state diagram information further specifies one or more stop states;

wherein said ~~programmatically~~ <u>automatically</u> generating comprises ~~programmatically~~ <u>automatically</u> generating graphical source code such that if during execution of the graphical program one of the stop states becomes active, then the graphical program is caused to stop execution.

20. (Currently Amended) The method of claim 1, further comprising:

receiving information specifying a change to the state diagram information;

~~programmatically~~ automatically updating the graphical program to reflect the specified change.

21. (Currently Amended) The method of claim 1,

wherein said ~~programmatically~~ automatically generating the graphical program comprises calling an application programming interface (API) enabling the ~~programmatic~~ automatic generation of a graphical program.

22. (Currently Amended) The method of claim 1,

wherein said ~~programmatically~~ automatically generating the graphical program comprises ~~programmatically~~ automatically requesting a server program to generate the graphical program.

23. (Currently Amended) A computer-implemented method for ~~programmatically~~ automatically generating a new graphical program, comprising:

receiving information specifying a state diagram, wherein the state diagram specifies first functionality;

executing a graphical program generation (GPG) program;

the GPG program ~~programmatically~~ automatically generating the new graphical program using said information, wherein the new graphical program includes graphical source code corresponding to the state diagram, wherein the new graphical program comprises a plurality of interconnected nodes which visually indicate operation of the graphical program, [[and]] wherein the new graphical program is executable by a computer to perform the first functionality, and wherein said ~~programmatically~~ automatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

24. (Cancelled)

25. (Currently Amended) A computer-implemented method for ~~programmatically~~ automatically generating a graphical program based on a state diagram, comprising:

displaying an initial state diagram;

~~programmatically~~ automatically generating a graphical program corresponding to the initial state diagram, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable by a computer, and wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating;

receiving user input specifying a change to the initial state diagram;

~~programmatically~~ automatically updating the graphical program to correspond to the specified change, in response to the user input specifying the change, wherein said automatically updating the graphical program updates the graphical program without any user input specifying the graphical program during said updating.

26. (Currently Amended) A system for ~~programmatically~~ automatically generating a graphical program, the system comprising:

a processor coupled to a memory, wherein the memory stores a graphical program generation (GPG) program;

wherein the processor is operable to execute the GPG program in order to:

receive state diagram information, wherein the state diagram information specifies one or more states;

~~programmatically~~ automatically generate the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality, and wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

9

27. (Cancelled)

28. (Currently Amended) The system of claim 26,

wherein said ~~programmatically~~ automatically generating the graphical program comprises ~~programmatically~~ automatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

29. (Currently Amended) A memory medium for ~~programmatically~~ automatically generating a graphical program based on a state diagram, the memory medium comprising program instructions executable to:

receive state diagram information, wherein the state diagram information specifies a plurality of states and one or more transitions between the plurality of states;

~~programmatically~~ automatically generate the graphical program in response to the state diagram information, wherein said ~~programmatically~~ automatically generating comprises ~~programmatically~~ automatically generating graphical source code corresponding to the specified one or more states, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate operation of the graphical program, [[and]] wherein the graphical program is executable by a computer to perform functionality specified by the state diagram information, and wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

30. (Cancelled)

31. (Currently Amended) The memory medium of claim 29,

wherein said ~~programmatically~~ automatically generating the graphical program comprises ~~programmatically~~ automatically generating a block diagram including the graphical source code corresponding to the specified ~~one or more~~ plurality of states.

10

32. (Currently Amended) A computer-implemented method for ~~programmatically~~ automatically generating a graphical program based on state diagram information, comprising:

receiving the state diagram information, wherein the state diagram information specifies a plurality of states and transitions between the states;

~~programmatically~~ automatically generating the graphical program in response to the state diagram information, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein a first one or more nodes comprise graphical source code executable to implement first functionality corresponding to a first one or more states, [[and]] wherein a second one or more nodes are user-configurable to implement second functionality of a corresponding second one or more states, and wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

33. (Previously Presented) The method of claim 32,

wherein the ~~programmatically~~ automatically generated graphical program includes placeholder graphical source code for each of the second one or more states.

34. (Previously Presented) The method of claim 33, further comprising:

for each of the second one or more nodes, a user manually entering graphical source code specifying execution instructions to be performed when the state is active during execution of the graphical program.

35. (Previously Presented) The method of claim 33,

wherein the placeholder graphical source code for each state comprises a case in a graphical case structure.

36. (New) A computer-implemented method for automatically generating a graphical program based on a state diagram, comprising:

11

receiving state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states;

automatically generating the graphical program in response to the state diagram information, wherein said automatically generating comprises automatically generating graphical source code corresponding to the plurality of states, wherein the graphical source code comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program is executable by a computer, and wherein said automatically generating the graphical program creates the graphical program without any user input selecting the nodes or establishing connections between the nodes.

37. (New) A memory medium comprising program instructions for automatically generating a graphical data flow program based on a state diagram, wherein the program instructions are executable by a processor to implement:

receiving state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states;

automatically generating the graphical data flow program in response to the state diagram information, wherein the graphical data flow program comprises a plurality of nodes and connections between the nodes, wherein the connections indicate that data produced by one node is used by another node, wherein the graphical data flow program is executable by a computer to implement functionality indicated by the state diagram information, and wherein said automatically generating the graphical data flow program creates the graphical data flow program without any user input specifying the graphical data flow program during said creating.

38. (New) The memory medium of claim 37, wherein said automatically generating the graphical data flow program creates the graphical data flow program without any user input selecting the nodes or creating the connections between the nodes.

39. (New) The memory medium of claim 37,

wherein said automatically generating the graphical data flow program comprises automatically generating a block diagram.

40. (New) The memory medium of claim 37,

wherein each of the plurality of nodes corresponds to one of the plurality of states.

41. (New) The memory medium of claim 37,

wherein, for at least one state of the plurality of states, the state diagram information specifies program code associated with the state;

wherein the automatically generated graphical data flow program includes the specified program code.

42. (New) The memory medium of claim 37,

wherein, for at least one state, the state diagram information specifies program code associated with the state;

wherein the automatically generated graphical data flow program is operable to invoke the specified source code.

43. (New) The memory medium of claim 37,

wherein the state diagram information further specifies one or more state transitions, wherein each state transition specifies a transition from a first state to a second state;

wherein said automatically generating further comprises automatically generating portions of the graphical data flow program corresponding to the specified state transitions.

44. (New) The memory medium of claim 37,

wherein the automatically generated graphical data flow program includes placeholder graphical source code for each state transition.

45. (New) The memory medium of claim 44,

wherein the state diagram information specifies at least two state transitions from a particular state;

wherein the state diagram information also specifies a priority ordering for the at least two state transitions;

wherein said automatically generating comprises automatically generating the graphical data flow program such that, during execution of the graphical data flow program, Boolean conditions associated with the at least two state transitions are evaluated in the specified priority ordering.

46. (New) The memory medium of claim 37,

wherein the state diagram information further specifies an initially active state;

wherein said automatically generating comprises automatically generating the graphical data flow program such that the graphical program begins execution in the initially active state.

47. (New) The memory medium of claim 37,

wherein the state diagram information further specifies one or more stop states;

wherein said automatically generating comprises automatically generating the graphical data flow program such that if during execution of the graphical data flow program one of the stop states becomes active, then the graphical data flow program is caused to stop execution.

48. (New) The memory medium of claim 37, wherein the program instructions are further executable to implement:

receiving information specifying a change to the state diagram information;

automatically updating the graphical data flow program to reflect the specified change.

49. (New) The memory medium of claim 37,

wherein the state diagram information comprises an executable program.

50. (New) A computer-implemented method for automatically generating a graphical data flow program based on a state diagram, the method comprising:

receiving state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states;

automatically generating the graphical data flow program in response to the state diagram information, wherein the graphical data flow program comprises a plurality of nodes and connections between the nodes, wherein the connections indicate that data produced by one node is used by another node, wherein the graphical program is executable by a computer to implement functionality indicated by the state diagram information, and wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

51. (New) A memory medium comprising program instructions for automatically generating a graphical data flow program based on a state diagram, wherein the program instructions are executable by a processor to implement:

receiving information that specifies a plurality of states, transitions between the states, and conditions for transitions between the states;

automatically generating the graphical data flow program in response to the information, wherein the graphical data flow program comprises a plurality of nodes and connections between the nodes, wherein the connections indicate that data produced by one node is used by another node, wherein the graphical data flow program is executable by a computer to implement functionality indicated by the information, and wherein said automatically generating the graphical data flow program creates the graphical data flow program without any user input specifying the graphical data flow program during said creating.

52. (New) The memory medium of claim 51, wherein said automatically generating the graphical data flow program creates the graphical data flow program without any user input selecting the nodes or creating the connections between the nodes.

53. (New) A memory medium comprising program instructions for automatically generating a graphical data flow program based on a state diagram, wherein the program instructions are executable by a processor to implement:

receiving information that specifies a plurality of states, transitions between the states, and start and stop states;

automatically generating the graphical data flow program in response to the information, wherein the graphical data flow program comprises a plurality of nodes and connections between the nodes, wherein the connections indicate that data produced by one node is used by another node, wherein the graphical data flow program is executable by a computer to implement functionality indicated by the information, and wherein said automatically generating the graphical data flow program creates the graphical data flow program without any user input specifying the graphical data flow program during said creating.

54. (New) The memory medium of claim 53, wherein said automatically generating the graphical data flow program creates the graphical data flow program without any user input selecting the nodes or creating the connections between the nodes.